CS101- Algorithms and Programming I

Lab 01

Lab Objectives: first Java programs, debugging

For all labs in CS 101, your solutions must conform to these CS101 style guidelines (rules!)

Step 1: Organize your H: drive.

- Organizing your work from the beginning is important. Set up the folder structure on your H: drive according to the instructions found at the following URL (http://www.cs.bilkent.edu.tr/~adayanik/cs101/practicalwork/defaultFolderSetup.htm)

Step 2: Practice compiling programs

- Read and follow these instructions to install and use DrJava.

Step 3: Debugging programs.

(part a)

- Create a project, Lab01a in the Lab01 folder on your H: drive.
- Download the file **Lab01a.java** and include it in this project. The program does not compile. Find and fix the syntax errors in the program so that it will compile and run. When it runs ***successfully*** the program will give the output shown. Check your output to ensure it appears correctly.

  Sample Run:

```
Hello everyone, below are some course details
Welcome to CS101 SPRING 2020 Lab 01
There are: 10 lab sessions in this course.
Labs contribute to gradePercent * 25% of your total grade.
This lab :  2.5   points
All labs :  25.0  points
Please come prepared...

Good luck!
```

Step 4: Creating Basic Programs

(part b)

According to a study published in the British Journal of Nutrition in 1991, if you are an adult, your percentage of body fat can be estimated as accurately as with skin-fold measurements and bioelectrical tests using the following gender-based formulas in conjunction with your BMI. Create a new project, **Lab01b**. Write a program that will determine the ideal body fat percentage (BFP) for men.

$$BFP = (1.20\,x\,BMI) + (0.23\,x\,age) - 16.2$$

- To calculate this, first you have to determine Body Mass Index (BMI) using the formula given below. The unit is kg/m^2.

$$BMI = weight / (height)^2$$

- Set the weight / height / age values as in sample run below, and store all values in variables. You don't need to input these values from the user. You should define CONSTANTS to store the necessary constant values. Remember to give constants and variables meaningful names.

- Calculate the BMI and BFP with the specified values and display them. Also, assuming the same age and the same height, by changing the weight to simulate weight gain/loss, recalculate the BMI and BFP, then display these values as shown in the sample run below on the console. For each BMI / BFP pair, you should give all output in one line.

Sample Run:

```
weight(kg): 72
height(m): 1.76
age(years): 25

Based on a height of 1.76 and weight of 72 and age of 25, your BMI is
23.243801652892564 and your BFP is %17.442561983471077

Based on a height of 1.76 and weight of 77 and age of 25 your BMI would
be 24.857954545454547 and your BFP would be 19.379545454545454%

Based on a height of 1.76 and weight of 67 and age of 25 your BMI would
be 21.62964876033058 and your BFP would be 15.505578512396696%
```

(part c)

While many people are familiar with the Fibonacci sequence (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, etc., where each number is the sum of the previous two numbers), few know there is a formula to figure out any given Fibonacci number: the formula that we have below, where F(n) is the nth Fibonacci number. That is, to find the 100th Fibonacci number, you don't have to calculate the first 99 numbers. You can just throw 100 into the formula.

$$F(n) = \frac{(\varphi)^n - (-\frac{1}{\varphi})^n}{\sqrt{5}}$$

where $\varphi = \frac{1+\sqrt{5}}{2}$. Note that this number is the Golden Ratio. Remarkably, even with all the square roots and divisions, the answer will be very close to a positive integer. Also, note that $F(0) = 0$ and $F(1) = 1$.

- Create a new project, **Lab01bc** in the Lab01 folder on your H: drive. Compute the given function above for an input n that you will set in the program. You don't need to input the value of n from the user.

  You should define CONSTANTS to store the necessary constant values. Remember to give constants and variables meaningful names.

  Sample Run 1 (setting n to 6):

  ```
  Fibonacci(6) is 8.000000000000002
  ```

  Sample Run 1 (setting n to 10):

  ```
  Fibonacci(10) is 55.000000000000014
  ```

  Sample Run 1 (setting n to 100):

  ```
  Fibonacci(100) is 3.542248481792631E20
  ```

  Sample Run 1 (setting n to 101):

  ```
  Fibonacci(101) is 5.73147844013819E20
  ```

- Since the Fibonacci numbers are integers, we need to throw away the tiny fractional parts in the results above. How can you do this? Display the correct integer result. Examine your integer results. Are they correct?